社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－MPS－51　(7)
2004／9／13

PC クラスタにおける混合整数計画問題の並列処理とその性能評価

田村慶一[†]　岩木稔[††]
高木允[†††]　北上始[†]

線形計画問題の一部の変数に対して整数制約を加えた問題を混合整数計画問題という．本論文では，分枝限定法と単体法を用いた混合整数計画問題解法の PC クラスタにおける並列化手法を提案する．混合整数計画問題の並列処理にはタスク分割によるマスタワーカモデルを使用する．並列化で課題となったのが負荷の偏りと総探索ノード数の増加である．負荷の偏りに関してはタスク奪い取りによる動的負荷分散手法を用い，総探索ノード数の増加に関しては暫定解の同期をおこない，課題を解決する．提案する並列化手法を実際に実装し，PC クラスタ上で性能評価をおこなった．性能評価により，提案する並列化手法で十分な台数効果が得られることを確認した．

Parallel Processing of Mixed Integer Programming Problem on PC Cluster and Its Performance Evaluation

Keiichi Tamura ,[†] Minoru Iwaki,[††] Makoto Takaki[†††]
and Hajime Kitakami[†]

A problem in which some variables of a linear programming problem can take only integer values and some variables can take fractional values is called a mixed-integer programming problem. The mixed-integer programming problem is solved using both the simplex method branch-and-bound algorithm. This paper proposes a parallelism of the mixed-integer programming problem on a PC cluster. The parallelism of the mixed-integer programming problem uses a task-divided-based master-worker model. There are two problems; inefficient load-balancing and increasing the total number of search nodes. To overcome the first problem, a task-steal-based dynamic load balancing technique is used for the dynamic load balancing of master-worker model. To solve the second problem, we propose synchronous techniques of incumbent. We implemented parallel mixed-integer programming problem on an actual PC clusters. The experimental results show good speed-up.

1. はじめに

線形計画問題の一部の変数に対して整数制約を加えた問題を混合整数計画問題[2] という．混合整数計画問題は，分枝限定法と単体法を用いた解法が存在するが，本論文はその解法の PC クラスタにおける並列化手法を提案する．

混合整数計画問題の並列処理にタスク分割によるマスタワーカモデルを適用すると，（1）各タスクの処理時間に極端な差があり十分なスピードアップを得られない，（2）暫定解をワーカプロセス間で同期させなければ総探索ノード数が増加する，という 2 つの点が課題となる．課題（1）を解決するために，タスク奪い取りによる動的負荷分散手法をマスタワーカモデルに適用する．課題（2）を解決するために，暫定解の同期方法を提案する．

提案する混合整数計画問題の並列化手法を PC クラスタ上で実装し，ジョブショップスケジューリング問題を使って性能評価をおこなった．性能評価の結果，提案する並列化手法を用いることにより，2 つの課題が解決できることを確認した．

† 広島市立大学情報科学部
Faculty of Information Sciences, Hiroshima City Unievrsity
†† NEC フィールディング株式会社
NEC Fielding,Ltd.
††† 広島市立大学大学院情報科学研究科
Graduate School of Information Sciences, Hiroshima City University

2. 混合整数計画問題

混合整数計画問題（P_0）は以下の問題形式で与えられる（（P_0）の連続緩和を（$\bar{P}_0$）とする）．

$$minimize\ z = c^t x + d^t y \quad (1)$$

$$subject\ to\ Ax + Ey = b \quad x \geq 0 \quad (2)$$

$$l^0 \leq y \leq u^0 \quad (3)$$

$$y \in Z^{n2} \quad (4)$$

まず，（$\bar{P}_0$）を単体法で解く．（$\bar{P}_0$）の最適解（$\bar{x}^0, \bar{y}^0$）とし，最適値を $\bar{z}^0$ とする．（$\bar{P}_0$）の最適解（$\bar{x}^0, \bar{y}^0$）が整数条件を満たしているならば，（$\bar{x}^0, \bar{y}^0$）は（P_0）の最適解となる．（$\bar{P}_0$）の最適解が整数条件

を満たしていないならば，y の中から 1 つの変数 y_s を選んで，部分問題（P_1）（P_2）を作成する．問題を部分問題に分割することを分枝操作という．

問題（P_k）の部分問題を（P_{k+1}）（P_{k+2}）とするとき，（P_{k+1}）（P_{k+2}）の最適解を，（x^{k+1}, y^{k+1}）と（x^{k+2}, y^{k+2}）とすると，最適値が小さい方が（P_k）の最適解となる．

部分問題の最適解のうち，最適値が最も良いものを暫定解（x^*, y^*）と呼ぶ（暫定解の目的値を暫定値 z^* と表す）．この暫定解と暫定値により，以下の場合分けをする．この操作を限定操作という．

（1） （$\bar{P}_k$）が最適解（$\bar{x}^k, \bar{y}^k$）を持ち，整数条件を満たすとき: 最適値 $\bar{z}^k$ が，現在得られている暫定値 z^* よりも良いならば，最適解（$\bar{x}^k, \bar{y}^k$）を新たな暫定解とする．

（2） （$\bar{P}_k$）が最適解（$\bar{x}^k, \bar{y}^k$）を持つが，整数条件を満たさないとき：最適値 $\bar{z}^k$ と暫定値 z^* とを比較し，もし $\bar{z}^k \leq z^*$ ならば，部分問題を生成する．

「分枝操作，分枝操作により生成された部分問題の連続緩和問題に対する単体法の実行，単体法による評価から限定操作をおこなう」ことを，これ以降，簡単に「分枝限定操作」と呼ぶことにする．分枝限定操作をおこなっていない部分問題を活性部分問題と呼ぶ．

どの活性部分問題から分枝限定操作をおこなうかを決定することを探索戦略という．本研究で採用している探索戦略は，最初の暫定解を発見するまで深さ優先探索でノードを選択し分枝限定操作をおこない，それ以降は幅優先探索でノードを選択する方法である．

変数の領域を分けながら部分問題を生成するのでは，条件が違うだけで単体法を毎回最初から解く必要がある．分枝限定法と単体法を使った混合計画問題の解法では，活性部分問題は，データ構造中に親問題を解いたときの，掃き出し処理の履歴，基底として選択されている列番号，非基底項の Bound 位置，基底項に対する最新の右辺項の値（元問題の b が単体法を実行していくうちに値が変化したもの）を持っている．以下，これらのデータを単体法実行履歴データと呼ぶ．

3. 並列処理の基本構造と課題

3.1 マスタワーカモデルの適用

マスタプロセスが持っている暫定解と暫定値をそれぞれグローバル暫定解とグローバル暫定値と呼ぶ．ワーカプロセスが持っている暫定解と暫定値をそれぞれローカル暫定解とローカル暫定値と呼ぶ．マスタプロセスとワーカプロセスの処理手順を示す．

マスタプロセス:

- フェーズ I:

（1） 最初の暫定解を発見するまで深さ優先探索でノードを選択し，選択したノードに対して分枝限定操作をおこなう．実行の過程で生成された活性部分問題をタスクとしてタスクプールに挿入する．各タスクには単体法履歴データが添付される．

（2） 最初の暫定解を発見すると，発見した暫定解と暫定値をそれぞれグローバル暫定解とグローバル暫定値に格納する．フェーズ II に移る．

- フェーズ II:

（1） サイト i のワーカプロセスよりタスクリクエストを受け取ると（2）に進む．

（2） タスクリクエストにはサイト i のワーカプロセスでの最新の暫定解と暫定値が添付されている．もし，グローバル暫定値よりも受け取ったローカル暫定値のほうが小さければ，グローバル暫定解とグローバル暫定値を更新する．

（3） タスクプールよりタスクを取り出す．選ばれるタスクはタスクが示すノードのうちルートノードに最も近いノードである．もし，タスクプールが空であれば終了処理に入る．

（4） （3）で取り出したタスク（単体法実行履歴データが添付されている）と，グローバル暫定解と，グローバル暫定値をサイト i のワーカプロセスに送信する．（1）に戻る．

ワーカプロセス:

（1） マスタプロセスにタスクリクエストを送信する．タスクリクエストにはローカル暫定解とローカル暫定値を添付する．

（2） マスタプロセスよりタスクを受け取ると，ローカル暫定値よりも受け取ったグローバル暫定値のほうが小さければローカル暫定解とローカル暫定値を更新する．マスタプロセスより終了通知を受け取ると処理を終了する．

（3） 受け取ったタスクが示すノードを起点とし分枝限定操作を繰り返していく．すべての分枝限定操作を終えると（1）に戻る．

3.2 課　　題

混合整数計画問題の並列処理にタスク分割によるマスタワーカ法を適用すると，以下の 2 つの課題が存在する．

- 負荷の偏り：
  混合整数計画問題では各タスクの負荷の大きさは一定ではない．極端に処理時間がかかるタスクが割り当てられたワーカプロセスの終了を待つこととなり，十分なスピードアップが得られない．

表 1 評価に使用した問題の各パラメータ

	逐次実行での処理時間	探索ノード数	ノードサイズ（byte）	整数変数の数	連続変数の数	制約式の数
4 仕事× 4 機械	0.229 (秒)	563	7220	30	20	80
5 仕事× 5 機械	238.814 （秒）	280033	11080	50	25	125
6 仕事× 5 機械	344.478 （秒）	115651	15780	75	30	180
6 仕事× 6 機械（1）	1070.481 （秒）	216825	18900	90	36	216
6 仕事× 6 機械（2）	74 (分)	469091	18900	90	36	216
7 仕事× 6 機械	23 （時間）	8498655	25548	126	42	294

- 総探索ノード数の増加：
  ワーカプロセスで得られた暫定解を他ワーカプロセスに伝えなければ逐次処理の時よりも分枝操作で生成されるノード数が増加する可能性がある．

4. 動的負荷分散手法

負荷の偏りを解消するために文献 3) で示されたタスク奪い取りによる動的負荷分散手法であるマスタ・タスク・ステイル法を用いる．単体法実行履歴データのデータサイズが大きいため，タスクの移動を最小限にすることができるマスタ・タスク・ステイル法は混合整数計画問題の並列化に有効な手法であるといえる．

タスクリクエスト受け取ったマスタプロセスは，もしタスクプールが空であればすべてのワーカプロセスにタスク奪い取り要求を出す．タスク奪い取り要求を受け取ったワーカプロセスは抱えている活性部分問題のうち，ルートノードに近い問題をマスタプロセスに送信する．マスタプロセスはワーカプロセスより活性部分問題が送信されてくると，それをタスクとしてタスクプールに挿入する．

5. 暫定解の同期

暫定解の同期をおこなうことにより，逐次実行のときよりも総探索ノード数が増加し最適な台数効果が得られなくなることを回避する．暫定解同期の頻度により，（a）ワーカプロセスよりタスクリクエストが届いた時，（b）タスク奪い取り時，（c）暫定解が見つかった時，に暫定解の同期を取るという 3 種類の同期方法を提案する．

6. 関 連 研 究

混合整数計画問題を分枝限定法と単体法で解くアルゴリズムはすでに文献 1) において共有メモリ型並列計算機上で並列化がおこなわれている．共有メモリ型並列計算機上での手法をそのまま PC クラスタ上に適用することはできない．

一方，分枝限定法のみの並列化に関する研究はその適用範囲が広く盛んに研究がおこなわれている．並列分枝限定法の研究で有効とされてきたのがマスタワーカモデルを分枝限定法に適用した並列分枝限定法[4] である．並列分枝限定法は，ワーカプロセスでおこなってよい分枝限定操作の深さを決め，生成されたタスクをマスタプロセスに一旦戻すことで，負荷の偏りと暫定解同期の問題を解決している．生成されたタスクをマスタプロセスに返すため，タスクが抱えるデータ量が多い混合整数計画問題に適用すると通信量の増加を招いてしまう．

7. 性 能 評 価

性能評価には，混合整数計画問題の応用例として頻繁に使用されるジョブショップスケジューリング問題を使用した．**表 1** に各問題の特徴である，逐次実行での処理時間，探索ノード数，ノードサイズ（byte），整数変数の数，連続変数の数と制約式の数を示す．

PC （CPU:Pentium4 2.53GHz, Memory:1.5GB, Disk:80GB）が 16 台， 100Mbit/sec イーサネットにつながっている PC クラスタを性能評価に使用した．

動的負荷分散手法と暫定解の同期方法で 5 種類に場合分けし測定をおこなった．場合（A）は暫定解の同期が（a），場合（B）は暫定解の同期が（c），場合（C）はタスク奪い取りがあり，暫定解の同期が（a），場合（D）はタスク奪い取りがあり，暫定解の同期が（b），場合（E）はタスク奪い取りがあり，暫定解の同期が（c），である．

7.1 台 数 効 果

場合（A）は**表 2**，場合（B）は**表 3**，場合（C）は**表 4**，場合（D）は**表 5**，場合（E）は**表 6** に測定結果を示す．表の値は逐次の処理時間を各台数で実行時に得られた処理時間で割った値であり，台数効果を示している．太字は超線形加速であることを示す．

5 仕事× 5 機械は，場合（C），場合（D）と場合（E）で超線形加速が得られた．5 仕事× 6 機械，6 仕事× 6 機械（1）もいくつか超線形加速が得られているものがあるが，いずれの結果も場合（E）が他の場合と比べて良い台数効果が得られている．

7.2 性能評価（動的負荷分散）

タスク奪い取りによる動的負荷分散手法が効果的に動いていることを確認するために各ワーカプロセスでの処理時間を測定した．紙面の制限上，6 仕事× 6 機械（1）での場合（B）と場合（E）の各ワーカプロセスでの処理時間の比較（**図 1**）を示す．場合（B）は，場合（E）と比べて処理時間に極端な差があることが分かる．

表 2 台数効果 - 場合（A） -

	2 台	4 台	8 台	16 台
4 仕事× 4 機械	0.92	0.81	0.64	0.80
5 仕事× 5 機械	1.77	2.80	2.80	2.88
6 仕事× 5 機械	1.41	2.04	2.03	2.09
6 仕事× 6 機械（1）	1.46	1.58	1.64	1.61

表 3 台数効果 - 場合（B） -

	2 台	4 台	8 台	16 台
4 仕事× 4 機械	0.94	0.95	0.88	0.83
5 仕事× 5 機械	1.73	2.88	2.82	2.81
6 仕事× 5 機械	1.54	2.12	2.09	2.13
6 仕事× 6 機械（1）	1.41	1.53	1.56	1.54

表 4 台数効果 - 場合（C） -

	2 台	4 台	8 台	16 台
4 仕事× 4 機械	0.77	0.66	0.64	0.25
5 仕事× 5 機械	**2.24**	**6.05**	**8.63**	**29.75**
6 仕事× 5 機械	1.69	3.05	6.85	9.91
6 仕事× 6 機械（1）	1.50	2.67	4.85	11.35

表 5 台数効果 - 場合（D） -

	2 台	4 台	8 台	16 台
4 仕事× 4 機械	0.83	0.71	0.40	0.36
5 仕事× 5 機械	**2.35**	**5.23**	**9.83**	**27.90**
6 仕事× 5 機械	1.71	**4.07**	6.89	11.80
6 仕事× 6 機械（1）	1.53	2.88	5.41	12.06

場合（E）は各サイトの処理時間が一定になっており，タスク奪い取りによる動的負荷分散手法の有効性を示すことができた．

7.3 性能評価（暫定解の同期）

6 仕事× 6 機械（1）において，場合（C）と，場合（D）と，場合（E）で総探索ノード数を比較した．**図 2** に測定結果を示す．暫定解の同期頻度は，場合（C）と場合（D）は，場合（E）と比べて小さいため，結果として総探索ノード数が増えている．

7.4 性能評価（大規模な問題）

大規模な問題を用いて場合（E）についての性能評価をおこなった（**表 7**）．7 仕事× 6 機械は逐次処理で約 1 日かかってしまうが，16 台で並列処理すると 1 時間以内で解くことができる．

8. ま と め

本論文では，PC クラスタにおける混合整数計画問題の並列処理とその性能評価を述べた．性能評価により並列化手法の有効性を示すことができた．これからの課題として性能モデルを立て提案手法の有効性を定式的に示すことと，グリッドコンピューティングへの展開を考えていきたい．

謝辞 本研究の一部は広島市立大学・特定研究費（一般研究費（コード番号： 3106））の支援により行われた．

表 6 台数効果 - 場合（E） -

	2 台	4 台	8 台	16 台
4 仕事× 4 機械	0.78	1.05	0.57	0.31
5 仕事× 5 機械	**2.68**	**11.61**	**18.81**	**25.58**
6 仕事× 5 機械	**2.55**	**4.56**	7.69	14.80
6 仕事× 6 機械（1）	**2.06**	**5.13**	7.14	14.91

表 7 大規模な問題での台数効果

	2 台	4 台	8 台	16 台
6 仕事× 6 機械（2）	**2.67**	**5.09**	**12.84**	**25.67**
7 仕事× 6 機械	**2.13**	**7.17**	**11.28**	**28.97**

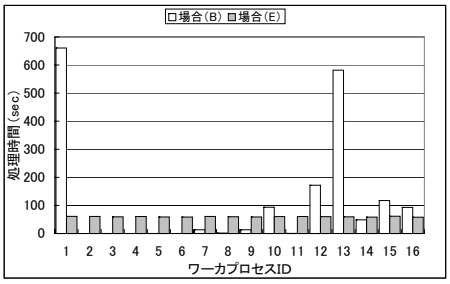

図 1 各サイトの処理時間（6 仕事× 6 機械（1））

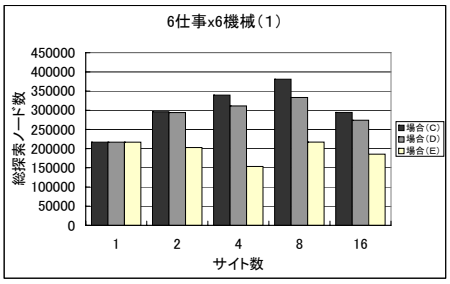

図 2 総探索ノード数（6 仕事× 6 機械（1））

参 考 文 献

1) H. Kitakami, H. Hara, H. Yamanaka, and T. Miyazaki. Performance evaluation for parallel mixed-integer linear programming system. *Optimization Methods and Software*, 3:257–272, 1994.

2) G. Mitra. Investigation of some branch and bound strategies for the solution of mixed integer linear programs. *Mathematical Programming*, 4:155–170, 1973.

3) M. TAKAKI, K. TAMURA, T. SUTOU, and H. KITAKAMI. Dynamic load balancing for parallel modified prefixspan. In *Proceedings of The 2004 International Conference on Parallel and Distributed Processing Techniques and Applications(PDPTA 2004)*, pp. 352–358. CSREA Press, 2004.

4) 中村, 山田, 二方, 合田. Pc クラスタ上での並列分枝限定法の高速化手法. 情報処理学会研究報告 HPC-95, 2003 年 8 月.